CLAIMS

What is claimed is:

Sub B1

1. A process for scheduling requests to access a resource, said requests originating from at least one thread from at least one initiator, said process comprising combining scheduling of requests between threads and scheduling of requests of initiator access to the resource and maintaining order of requests within each thread.

2. The process as set forth in claim 1, wherein combining comprises using a combination of thread quality of service (QOS) scheduling and resource scheduling.

3. The process as set forth in claim 2, wherein combining further comprises:

determining an order of requests to meet QOS guarantees;

determining an order of requests for resource efficiency; and

if the resource efficiency order satisfies QOS guarantees, and intra-thread order is maintained, scheduling a request according to a first resource efficiency order, else scheduling a request in accordance with a second resource efficiency order.

4. The process as set forth in claim 1, further comprising maintaining and using a thread scheduling history to at least in part determine scheduling of threads.

5. The process as set forth in claim 4, wherein thread scheduling history comprises thread bandwidth usage.

6. The process as set forth in claim 1, further comprising maintaining a state and access history on the device to at least in part determine scheduling of the resource.

7. The process as set forth in claim 1, wherein scheduling is determined by prioritizing threads according to bandwidth usage and sequencing requests from different threads so as to achieve a determined device performance.

8. The process as set forth in claim 1, wherein scheduling is selected from the group consisting of absolute and cost-function scheduling.

9. The process as set forth in claim 1, wherein the resource is a dynamic random access memory (DRAM) and scheduling is selected from the group consisting of deciding when to close dynamic random access page (DRAM) and open another, and deciding when to switch DRAM requests to use a different physical bank of DRAM, and deciding when to switch direction of a bus coupled to the DRAM.

10. A scheduling apparatus for scheduling access to a resource, comprising:

an input coupled to receive at least one access request originating from at least one thread from at least one initiator;

logic to combine scheduling of requests between threads and scheduling of initiator access to the resource and maintaining order of requests within each thread.

11. The scheduling apparatus as set forth in claim 10, wherein the resource is selected from the group consisting of a process, apparatus and a dynamic random access memory (DRAM).

12. The scheduling apparatus as set forth in claim 10, wherein the logic utilizes a combination of thread quality of service (QOS) guarantees and resource cost-function scheduling.

13. The scheduling apparatus as set forth in claim 10, further comprising a thread scheduling history, said logic using the thread history to at least in part determine scheduling.

14. The scheduling apparatus as set forth in claim 13, wherein the thread scheduling history comprises thread bandwidth usage.

15. The scheduling apparatus as set forth in claim 10, further comprising a state and access history used to at least in part determine scheduling of the resource.

16. The scheduling apparatus as set forth in claim 10, wherein scheduling of threads from at least one initiator is selected from the group consisting of absolute and cost function scheduling.

17. The scheduling apparatus as set forth in claim 10, wherein the resource is dynamic random access memory (DRAM) and a cost function scheduling is selected from the group consisting of deciding when to close a DRAM page and open another, deciding when to switch DRAM requests to use a different physical band of DRAM, and deciding when to switch direction of a bus coupled to the DRAM.